

Mini-Howto Servidores de Alta disponibilidad (*heartbeat + drbd*)

Heartbeat

1- Que es y que hace heartbeat

Bien, en principio vamos a ver el funcionamiento de este sistema de alta disponibilidad...

El software heartbeat trabaja enviando latidos (ping), los cuales verifican si el Server principal esta activo o no, estos pings enviados por heartbeat requieren una respuesta por parte del Server principal o master, si al cabo de un cierto tiempo el Server no responde dichos ping, heartbeat determina que ese Server se encuentra inactivo/caído, y automáticamente activa al Server secundario para que asuma el control de la red, en nuestro caso o ejemplo que vamos a usar este asumirá el control de la red usando el IP 192.168.0.88/32. Con este funcionamiento tenemos que el administrador de red, puede estar tranquilo, ya que en caso de que ocurra un problema con el Server principal el Server secundario asumirá el control automáticamente, luego uno tendrá que trabajar para volver a colocar el linea el Server primario/caído por la falla que sea, pero lo bueno es que uno lo puede realizar totalmente tranquilo a dicho trabajo, ya que el secundario nos esta dando una mano y haciendo todo lo que el que el Server principal tendría que estar haciendo, con esto lo que logramos es una alta disponibilidad de servicios, como veremos en este caso estaremos dando alta disponibilidad de conexión a Internet, servidor web, y datos... porque digo de datos, porque mas adelante explicaremos como hacer funcionar drbd sistema de raid1 en red.

2- Instalación y puesta en marcha

Primero que todo necesitamos 2 PC, cada una como mínimo 2 placas de red. acordemos nos que una es destinada para el latido de heartbeat.

Los paquetes para woody los podemos encontrar en....

http://www.ultramonkey.org/download/heartbeat/1.2.2/debian_woody/

Para otras plataformas podrían encontrar los rpm, fuentes, en google..

Los paquetes necesario son:

```
heartbeat-dev_1.2.2-10woody_i386.deb
heartbeat_1.2.2-10woody_i386.deb
ldirectord_1.2.2-10woody_all.deb
libpils-dev_1.2.2-10woody_i386.deb
libpils0_1.2.2-10woody_i386.deb
libstonith-dev_1.2.2-10woody_i386.deb
libstonith0_1.2.2-10woody_i386.deb
stonith_1.2.2-10woody_i386.deb
```

Pagina WEB donde encontrarlos:

<http://www.peredanet.com.ar/utiles/ha/debs-heartbeat/>

Bueno una vez instalados estos paquetes, ya sea bajo Debian o cualquier otra disto tendremos que empezar a configurar el heartbeat, ahí vamos.

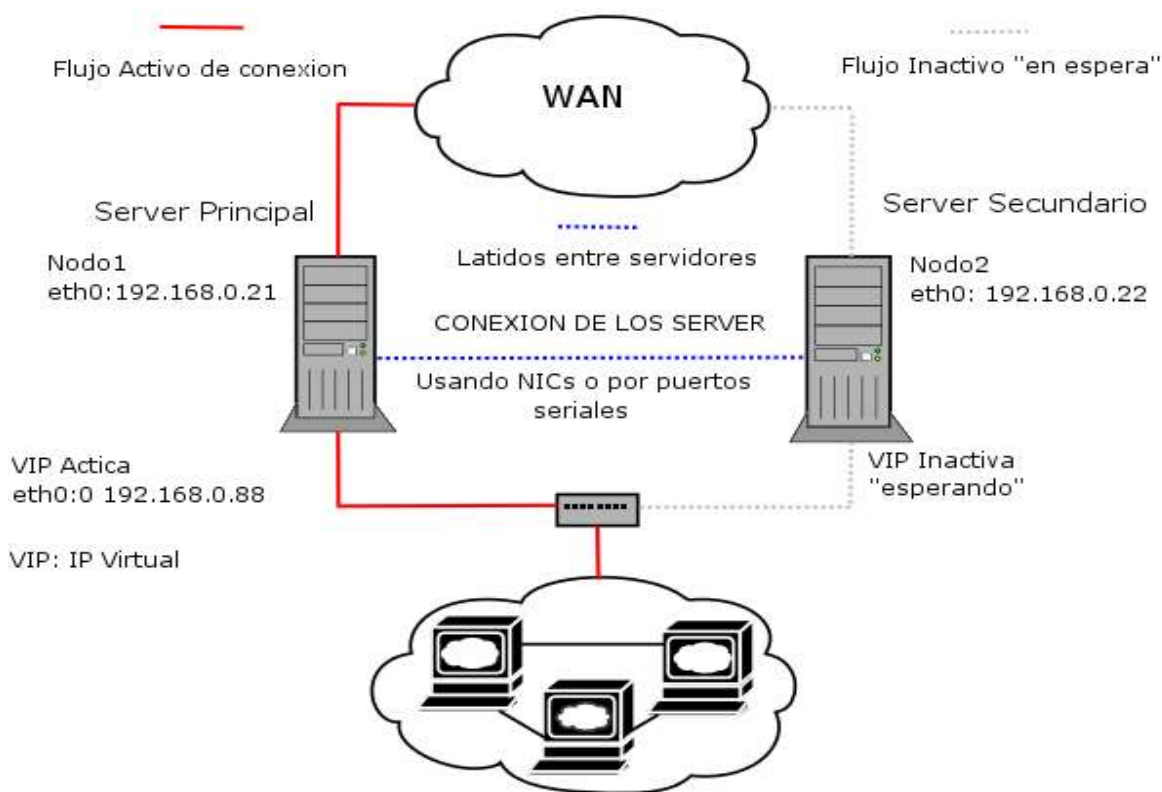
Primero definimos los nombres a cada PC miembro de cluster de HA en este caso vamos a usar 2 Pc a la que llamaremos nodo1 "primaria" y nodo2 "secundaria"

Arquitectura de las PC:

- Nodo1 "cyrix PR233 con 64Mb de Ram, disco de 2Gb"
- Nodo2 "Celeron 333 con 32Mb de Ram, disco de 4,3Gb"

Una vez que tenemos ambos Server ya listos, los conectaremos en red usando NIC's (Network interface Card) tarjetas de red, o bien por los puertos seriales (no lo probé). Aca les muestro un esquema básico de como quedaría la conexión en red de ambos nodos/Server:

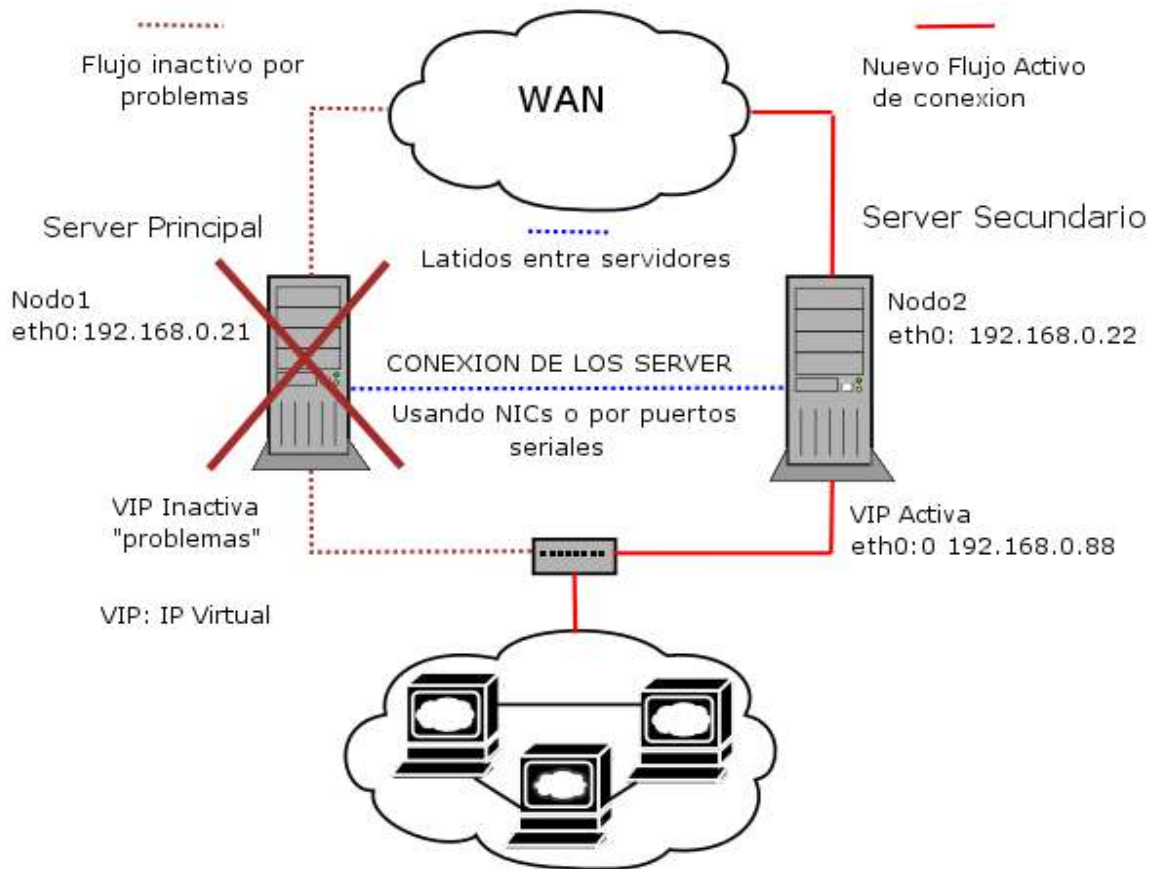
FUNCIONAMIENTO NORMAL



Este es el funcionamiento normal con el que funciona la alta disponibilidad, como vemos ahí el Nodo1 es quien en este momento posee la IP virtual vemos como el server secundario "nodo1" esta en modo de espera, siempre atento a que el nodo1 este funcionando adecuadamente, para eso utiliza el latido.

Ahora vemos un ejemplo si se corta el enlace del nodo1 o sufre algún problema de hardware o software grave.

FUNCIONAMIENTO EN MODO BACKUP "HA"



Como dijimos con anterioridad este es el esquema que realiza heartbeat cuando entra en funcionamiento el nodo2. Así es como tendría que andar cuando rayamos terminado de leer dicho howto

Bueno, para que todo quede funcionando según los gráficos tendremos que poner manos a la obra

Anteriormente les comentaba que tenemos que definir los nombres de las PC, en mi caso las agregué al DNS server que tengo en mi casa y el en /etc/hosts de cada nodo.

Por ejemplo les muestro el /etc/hosts del "nodo2"

```
nodo2:~# cat /etc/hosts
127.0.0.1    localhost
192.168.0.21 nodo1
192.168.0.22 nodo2
```

bueno en el nodo 1 tendría que estar de la misma forma.

3-Configuración

Ahora es momento de meternos con la configuración de heartbeat, prácticamente tendremos que tocar/jugar con 3 archivos de heartbeat los cuales se encuentran en /etc/ha.d/ ...

/etc/ha.d/ha.cf En este configuraremos el modo de conexión de ambos equipos

/etc/ha.d/haresources Este es el archivo que configuraremos los servicios que serían activados en el Server secundario en caso que el Server principal sufra algún conflicto.

/etc/ha.d/authkeys Justamente este archivo es el encargado de la seguridad del sistema, en lo referente a encriptación de los paquetes transmitidos, como también del sistema en si, verificación de archivos alterados, etc.

Empezamos mirando el primer archivo.

vi /etc/ha.d/ha.cf

```
-----
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
#####
#####
#
#   keepalive: fija el tiempo entre latidos " 2 segundos"
keepalive 2
#
#####
#####
#   deadtime: El nodo esta muerto después de 10 segundos
deadtime 10
#
#####
#####
#   warntime: Segundos antes de publicar el ultimo latidos, advierte a los registros
warntime 3
#####
#####
#initdead: para que no empiece a funcionar apenas arranque la PC, esto es depende
#cuanto se demore para cargar todos los servicios al arrancar, recomendable 3 veces
#o mas del deadtime
initdead 100
#####
#####
#   hopfudge: opción para redes en anillos " números de saltos "
#hopfudge 1
#
#####
#####
#   serial: Para usar un canal alternativo a de red "latidos entre nodos"
#serial /dev/ttyS0
#
#####
```

```
#####
# Baud: rate for serial ports...
baud 19200
#####
#####
#
# udpport: puerto que usa para el latido
udpport 694
#####
#####
# Que interfaz vamos a usar para el latido
bcast eth1
#####
#####
# watchdog: por si queremos usar watchdog en conjunto con heartbeat #"recomendado"
watchdog /dev/watchdog
# auto_failback: si queremos o no restablecer el orden de nodo1 "Primario" y #nodo2
"Secundario"

auto_failback on

respawn hacluster /usr/lib/heartbeat/ipfail

#ping lugmen.org.ar
#si el nodo1 pierde conexión con el LUGMen se activa el nodo2 saliendo por el otro #ISP :P

ping_group name linux-ha.org lugmen.org.ar
#Si pierde conexión con los 2 hace los mismo que el anterior "mas seguro" ya que uno de los
server a los que #estamos haciendo ping puede ser quien sufra de conectividad y no nuestro
nodo, cuando no tiene conectividad #con ninguno de los 2 seguramente si es problema nuestro
(normalmente)

#
#####
#####
# node nodename ... -- must match uname -n
node nodo1
node nodo2
-----
```

Fin del archivo... **(este como los otros dos archivos de la configuración de HA tendrán que ser idénticos en todos los miembros del "cluster "en nuestro caso nodo1 y nodo2)**

ahora veamos el siguiente archivo.

```
-----
# vi /etc/ha.d/haresources
-----
#
# This is a list of resources that move from machine to machine as
# nodes go down and come up in the cluster. Do not include
# "administrative" or fixed IP addresses in this file.
#
# We refer to this file when we're coming up, and when a machine is being
# taken over after going down.
#
# You need to make this right for your installation, then install it in
```

```

# /etc/ha.d
#
# These resources in this file are either IP addresses, or the name
# of scripts to run to "start" or "stop" the given resource.
#
# The format is like this:
#
#node-name resource1 resource2 ... resourceN

#nodo1 192.168.0.88 apache #- -> si no usamos DRBD

nodo1 drbddisk::drbd0 Filesystem::/dev/nb0::/home::reiserfs 192.168.0.88 apache #- -> usando
DRBD 0.7.0

# aclaramos que el cluster siempre responderá con esa IP... es decir esa va a ser
#la dirección IP que va a ser virtual en el cluster, y sera usada depende al caso #por el nodo 1 o 2
#un dato extra es que cada servicio va separado por espacios, en debian para realizar alta
disponibilidad de #apache tendremos que crear el enlace simbólico correspondiente.
#ln -s /etc/init.d/apache /etc/ha.d/resource.d/apache
# así con otros servicios que queramos

```

Cómo se ha explicado con anterioridad, el archivo `/etc/ha.d/haresources` especifica en primer lugar cual de las dos máquinas es la principal (***las dos máquinas deben de tener este archivo idéntico ya que han de saber en todo momento cual es la máquina destinada como principal y cual como secundaria***). En segundo lugar, especifica la IP que ha de soportar el servicio de red de alta disponibilidad. Por último, especifica cuales han de ser estos servicios.

!!Atentos a esto, ahí es donde ponemos el Número IP que sera el virtual o el que se pondrá cada server según este, este o no activo, no hay que poner esta dirección ni en /etc/network/interfaces ni nada.. atentos con eso!!

y el tercero es el siguiente

```

-----
# vi /etc/ha.d/authkeys
-----
#
# Authentication file. Must be mode 600
#
#3 md5 Hello!

auth 1
1 crc

# everything beyond this point is preserved

```

(también tiene que ser idéntico el archivo entre los nodos)

Como vemos en este caso estamos usando la placa eth1 para el latido, con ip 10.10.10.1 el nodo1 y 10.10.10.2 el nodo2.

La IP virtual que estamos usando y que sera trasportada de nodo a nodo en caso de falla de alguno de los mismos sera la IP 192.168.0.88 ...

Cuando cambia el servidor por problemas de uno de ellos es necesario que todas las máquinas incluso el router actualices su cache ARP, La actualización de las tablas se consigue mediante el envío de peticiones y respuestas ARP a la fuerza "gratuitous ARP" este es llevado a cabo por el código IPAddr el cual es ejecutado por heartbeat cuando se produce un fallo en la máquina el servicio "failover" o bien cuando el server principal vuelve a estar vivo "failback".

Ejemplo de como el nodo secundario se hace cargo del IP con el que queremos realizar alta disponibilidad como los servicios

```
heartbeat: 2004/08/03_08:19:56 info: Link nodo1:eth1 dead.
heartbeat: 2004/08/03_08:19:56 info: Running /etc/ha.d/rc.d/status status
heartbeat: 2004/08/03_08:19:56 info: No local resources [/usr/lib/heartbeat/ResourceManager
listkeys nodo2] to acquire.
heartbeat: 2004/08/03_08:19:57 info: Taking over resource group drbddisk::drbd0
heartbeat: 2004/08/03_08:19:57 info: Acquiring resource group: nodo1 drbddisk::drbd0
Filesystem::/dev/nb0::/home::reiserfs 192.168.0.88 apache
heartbeat: 2004/08/03_08:19:57 info: Running /etc/ha.d/resource.d/drbdisk drbd0 start
heartbeat: 2004/08/03_08:19:57 info: Running /etc/ha.d/resource.d/Filesystem /dev/nb0 /home
reiserfs start
heartbeat: 2004/08/03_08:19:58 info: Running /etc/ha.d/resource.d/IPAddr 192.168.0.88 start
heartbeat: 2004/08/03_08:19:58 WARN: Late heartbeat: Node lugmen.org.ar: interval 6060 ms
heartbeat: 2004/08/03_08:19:58 info: /sbin/ifconfig eth0:0 192.168.0.88 netmask 255.255.255.0
broadcast 192.168.0.255
heartbeat: 2004/08/03_08:19:58 info: for 192.168.0.88 on eth0:0 [eth0]
heartbeat: 2004/08/03_08:19:58 Sending Gratuitous Arp /usr/lib/heartbeat/send_arp -i 1010 -r 5
-p /var/lib/heartbeat/rsctmp/send_arp/send_arp-192.168.0.88 eth0 192.168.0.88 auto 192.168.0.88
fffffffffff
heartbeat: 2004/08/03_08:19:59 info: Running /etc/ha.d/resource.d/apache start
heartbeat: 2004/08/03_08:20:01 info: /usr/lib/heartbeat/mach_down: nice_failback: foreign
resources acquired
heartbeat: 2004/08/03_08:20:01 info: mach_down takeover complete.
heartbeat: 2004/08/03_08:20:01 info: mach_down takeover complete for node nodo1.
```

Como vemos este es el log de heartbeat del nodo 2.. vemos que detecta que el nodo1 murió entonces hace un manos a lo obra ;) , adquiere el grupo de lo que ponemos en el /etc/ha.d/haresources y empieza a levantar la ip, una vez que tiene la ip con la NIC correspondiente up (en nuestro caso eth0:0) vemos como manda un Sending Gratuitous Arp para que todos conozcan su nueva dirección MAC, y después levanta el servicio o servicios que queremos dar alta disponibilidad.

Nótese que en ningún momento se este indicando a heartbeat por que interfaz de red se han de ofrecer los servicios de alta disponibilidad. Es decir, se le dice que IP es la que ha de dar estos servicios pero no la interfaz que ha de ser configurada con esa IP. En el caso de que la máquina disponga de mas de una tarjeta de red, heartbeat recurre a la tabla de enrutado del sistema para decidir que interfaz de red se hará cargo. Por ejemplo, uno podría pensar que si el servicio de alta disponibilidad se ha de dar por la 192.168.0.88, añadir esta entrada a la tabla de rutado con: #route add 192.168.0.21 eth0 o ifconfig 192.168.0.21 eth0 sería suficiente para que heartbeat supiera que la IP 192.168.0.88 ha de ser levantada utilizando la interfaz eth0 creando un alias automático.

```
eth0:0 Link encap:Ethernet HWaddr 00:E0:7D:ED:43:76
inet addr:192.168.0.88 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Interrupt:9 Base address:0xef00
```

Aca llegamos al fin de la puesta en marcha del Heartbeat por lo que podemos empezar a descargar el drbd :P

FIN.

4- DRBD

4.1- Que es DRBD ??

Drbd es un dispositivo de bloques que está diseñado para construir "clusters" de alta disponibilidad. Esto es hecho por medio de una copia idéntica de todo un dispositivo de bloques a través de una red (dedicada en lo posible). Esto podría ser visto como un RAID-1 de red. :P

¿Cual es el alcance de drbd?, ¿Qué más necesito para construir un "cluster" de Alta Disponibilidad?

Drbd toma los datos, los escribe en el disco local y los envía al otro "host". En el otro "host", éste pone los datos en el disco.

Entre los otros componentes necesarios, está un servicio de pertenencia a un "cluster", que puede ser heartbeat, y algún tipo de aplicación que trabaje en un nivel más alto que un dispositivo de bloques.

Por ejemplo:

Un sistema de archivos & fsck.

Un sistema de archivos con "journaling"

Una base de datos con capacidades de recuperación.

(sacado de www.drbd.org ---> Mejor que ellos no lo voy a explicar :D)

4.2- Descargando DRBD

Bueno, ahora veamos que necesitamos para instalar drbd, en principio vamos a tener que descargar el paquete yo estoy probando la versión drbd-0.7.0, pasando anteriormente por la versión 0.6.12.

Descargamos las fuentes... o uno puede agregar en su sources.list los siguientes repositorios

```
deb http://fsrc.csee.wvu.edu/debian/apt-repository binary/  
deb-src http://fsrc.csee.wvu.edu/debian/apt-repository source/
```

Yo uso debian pero en este caso opte por la compilacion "compileidi"

```
#wget http://www.peredanet.com.ar/utiles/ha/drbd-0.7.0.tar.gz
```

4.3- Compilando Drbd

Donde descargamos el fuente hacemos.

```
# tar xvzf drbd-0.7.0.tar.gz  
# cd drbd-0.7.0  
# make clean  
# make  
Instale el módulo  
# make install
```

Ahora en teoria esta instalado el modulo del kernel, para esto nos tenemos que acordar de que contenemos las fuentes de kernel en /usr/src/linux

El archivo principal de drbd es el archivo /etc/drbd.conf

antes que nada, cargamos el modulo en los 2 nodos

```
#modprobe drbd
```

y nos fijamos.

```
#lsmod
```

5 Configurando

Pongamos los ejemplos con los que yo lo estoy usando.

Tenemos nodo1 "primario" con Dirección IP 192.168.0.21 y nodo2 "secundario" con dirección IP 192.168.0.22, Las particiones físicas son el el nodo1 /dev/hda3 y en el nodo2 /dev/hda4, pueden ser cualquiera la partición, en lo posible que seas del mismo tamaño ya que sino vamos a desperdiciar lo que sobre de la mas grande "RAID1".

Bueno una vez que tenemos las particiones con su sistema de archivos definido hacemos lo siguiente.

En el nodo2:

```
#drbdsetup /dev/nb0 disk /dev/hda4 /dev/hda1 0  
#drbdsetup /dev/nb0 net 192.168.0.22 192.168.0.21 C
```

El /dev/hda1 sera el metadisk, este lo que hace es ocupar 256Mb de espacio en disco para realizar un sistema de quien es el Server que esta mas actualizado, y no tener que hacer un syncall cuando el nodo1 vuelva de estar caído.

En el Nodo1:

```
#drbdsetup /dev/nb0 disk /dev/hda3 /dev/hda1 0  
#drbdsetup /dev/nb0 net 192.168.0.21 192.168.0.22 C  
#drbdsetup /dev/nb0 primary
```

Como vemos, le asignamos el protocolo C.. y al nodo1 como primario

Porque decimos Protocolo C? cuantas hay?

Tabla 1. Protocolos de DRBD

Protocolo	Descripción
A	La operación es completa cuando se escriban al disco y se envíen a la red.
B	La operación es completa cuando llegue un reconocimiento de la recepción.
C	La operación es completa cuando llegue un reconocimiento de que escribió ok.

Usamos la C porque es la mas segura, también un poco mas lenta.

Ahora Editamos el archivo que les comente anteriormente, este tiene que ser idéntico en los 2 o mas nodos.

```
-----  
#vi /etc/drbd.conf  
-----
```

```
resource drbd0 {  
  
    protocol C;
```

```

incon-degr-cmd "halt -f";

startup {
wfc-timeout 45; # wait for 45 seconds for connection otherwise continue
degr-wfc-timeout 30; # 30 seconds
}

# Si detecta errores en el disco...
disk {
on-io-error panic;
}

#Opciones de red, velocidad, etc
syncer {
rate 10M;
group 1;
al-extents 257;
}
# Numero/nombre del nodo
on nodo1 {
# Dispositivo del drbd
device /dev/nb0;
# Dispositivo físico para dicho nodo
disk /dev/hda3;
# Ponemos IP y puerto del nodo
address 192.168.0.21:7788;
# El metadata disk "intercambio" [index]
meta-disk /dev/hda1[0];
}

# Numero/nombre del nodo
on nodo2 {
# Dispositivo del drb
device /dev/nb0;
# Dispositivo físico para dicho nodo
disk /dev/hda2;
# Ponemos IP y puerto del nodo
address 192.168.0.22:7788;
# El metadata disk "intercambio" [index]
meta-disk /dev/hda1[0];
}
}
}

```

Lo que nos falta hacer en el Server primario es:

```
mount /dev/nb0 /home
```

¡No montar nunca el /dev/nb0 en el nodo2 cuando el nodo1 esta up!
 “hay que dejar que de eso se encargue heartbeat”

Antes la partición /home que es parte del dispositivo físico /dev/hda3 tiene que tener un fail system, se puede hacer antes o una vez hecho el dispositivo nb0 podemos hacer

```
mkfsreiserfs /dev/nb0
```

Bueno ahora tenemos que editar el fstab con la siguiente información.
En mi caso estoy usando el sistema de archivos reiserfs.

En los dos nodos poner lo siguiente "no tiene que estar automático al inicio en ninguno de ellos"
eso va a ser manejado por heartbeat + drbd

```
/dev/nb0 /home reiserfs defaults,noauto 0 0
```

En este caso como vemos, estoy haciendo duplicación de datos de /home/

Una vez que tenemos todo esto configurado nos disponemos a arrancar el drbd.

```
#/etc/init.d/drbd start (en los 2 nodos) El drbd del primario no va a arrancar hasta que este up el  
drbd secundario. "por lo menos creo que es así"
```

para ver si se esta sincronización o para ver si esta funcionando podemos hacer un

```
#watch -n1 cat /proc/drbd  
en los respectivos nodos.
```

6- Algunas Preguntas

- 1-Que es lo que pasa si se cae el primario?
- 2-Que pasa si cae el secundario?
- 3-Que pasa cuando vuelve a la vida el primario?

6.1- Algunas Respuestas

- 1- El secundario toma el control con los datos sincronizados antes que el primario caiga
- 2- Si el primario vive y el secundario muere, cuando resucita el secundario se hace Quick. "solo actualiza"
- 3- Si el primario muere y el secundario vive, cuando resucita el primario utiliza lo que nos encontramos como data disk para ponerse al tanto de solo los cambios hasta desde que el dejo de dar servicio.

La versiones anteriores este paso se tenia que hacer a mano de la siguiente forma:

- El primario debes resucitarlo como si fuera secundario usando dbrdsetup apropiadamente. De ese modo tienes `_dos_ secundarios` y "ningún primario". El "nuevo" secundario hará un Quick "upgrade" contra el secundario "viejo". Una vez terminado, manualmente conviertes el "nuevo" secundario en primario.

7- Compatibilidad

Drbd trabaja bien con particiones IDE y SCSI y unidades completas. No trabaja sobre el dispositivo de bloques del bucle. (Si lo trata, se bloqueará)

Drbd tampoco trabaja con el dispositivo de red de "loop-back".
(También observará cómo se bloquea: todas las peticiones son ocupadas por el dispositivo de envío y el proceso de envío está bloqueado en `sock_sendmsg()`. El hilo de recepción obtiene un bloque de la red y trata de ponerla en el caché, pero desafortunadamente el sistema debe decidirse por pasar algunos bloques del caché al disco. Esto ocurre en el contexto de recepción, y cuando todas las peticiones son ocupadas el receptor se bloquea.)

::Dichos del autor de drbd::

Referencias

<http://www.linux-ha.org/heartbeat/> (pagina oficial de heartbeat)

<http://www.drbd.org/> (pagina oficial de drbd)

<http://www.google.com.ar/linux> (google rules)

Gracias a la colaboración de:

Juan Eduardo Vitale (por correcciones en el howto y por la compañía de algunas tardes de la investigación)

Kastor (por correcciones y sugerencias hacia el howto)

Información recopilada

Federico Esteban Pereda

(pisa@lugmen.org.ar)

id jabber: pisa@lugmen.org.ar